#6
10/9/03

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appl.No.:	09/494,218	Confirmation No.: 3161
Appellant:	Brewer	
Filed:	January 28, 2000	
TC/A.U.	2142	
Examiner:	Blair	

Docket No.: TI-28385
Cust.No.: 23494

APPELLANT'S BRIEF (in triplicate)

Commissioner for Patents
P.O.Box 1450
Alexandria VA 22313-1450

Sir:

The attached sheets contain the Rule 192(c) items of appellant's brief. The Commissioner is hereby authorized to charge the fee for filing a brief in support of the appeal plus any other necessary fees to the deposit account of Texas Instruments Incorporated, account No. 20-0668; two additional copies of this first sheet of appellants' brief are enclosed.

Respectfully submitted,

Carlton H. Hoel
Reg. No. 29,934
Texas Instruments Incorporated
PO Box 655474, M/S 3999
Dallas, Texas 75265
972 917 4365

10/10/2003 SFORD1 00000002 200668 09494218
01 FC:1402 330.00 DA

Rule 192(c)(1) Real party of interest

Texas Instruments Incorporated owns the application.

Rule 192(c)(2) Related appeals and interferences

There are no related dispositive appeals or interferences.

Rule 192(c)(3) Status of claims

Claims 1-9 are pending in the application with all claims finally rejected. This appeal involves the finally rejected claims.

Rule 192(c)(4) Status of amendments

There is no amendment after final rejection.

Rule 192(c)(5) Summary of the invention

The invention provides a method and apparatus for transferring Java files from a server to a client by use of an intermediate gateway which determines portions of the Java files already transferred to the client so that only updates need by transferred. Application Fig.3 illustrates the apparatus and also notes that the transmission from the gateway to the client may be by a wireless link; so limiting the amount of information transferred is an advantage.

Rule 192(c)(6) Issues

The issues presented on appeal are:

(1) whether claim 5 is anticipated by the Brown reference.

(2) whether claims 1-3, 6-7, and 9 are patentable over the Brown reference in view of the Mishra reference.

(3) whether claim 4 is patentable over the Brown reference in view of the Mishra and Arnold references.

(4) whether claim 8 is patentable over the Brown reference in view of the Arnold reference.

(5) whether claims 2, 6-7, and 9 are supported by the specification.

Rule 192(c)(7) Grouping of the claims

The claims are grouped the same as the rejections.

Rule 192(c)(8) Argument

(1) Claim 5 was rejected as anticipated by Brown. The Examiner cited column 7, lines 25-44 for the gateway.

Appellant replies that Brown column 7, lines 25-44 describes creation of FCCFs (fully caffeinated class files) in a server which are delivered directly to a client as illustrated in Brown Fig.1; there is no suggestion of a gateway between the server and the clients as required by claim 5. The Examiner's assertion that the gateway could be a compiler in the server ignores the claim language.

(2) Claims 1-3, 6-7, and 9 were rejected as unpatentable over Brown in view of Mishra. The Examiner added Mishra to show sending only new portions of a file.

Appellant repeats the foregoing argument (1) regarding the gateway not appeating in or suggested by Brown.

(3) Claim 4 was rejected as unpatentable over Brown in view of Mishra and Arnold.

Appellant again repeats the foregoing argument (1) regarding the gateway not appearing in or suggested by Brown.

(4) Claim 8 was rejected as unpatentable over Brown in view of Arnold.

Appellant further repeats the foregoing argument (1) regarding the gateway not appearing in or suggested by Brown.

Consequently, the claims are patentable over the references.

(5) Claims 2, 6-7, and 9 were rejected as not supported by the specification with regard to the new portion of a file.

Appellant replies application Fig.3 and description page 6, bottom to page 7, top note that the gateway has a memory and sends files to the client, so the gateway impliedly can determine the new portion.

Rule 192(c)(9) Appendix

1. A method for loading class files from a server to a client comprising:

loading an application class onto a gateway server that preloads and preresolves said class;

creating a binary representation of new portions of the preloaded and preresolved class at said gateway; and

sending only the new portion to the client.

2. A method for loading Java class files from a server to a client device comprising the steps of:

a. gateway retrieving a Java class file;

b. gateway preloading and preresolving said Java class file and creating a representation of the Java class file;

c. determining at the gateway new portions of said representation of the Java class file not loaded in said client device;

d. creating at the gateway a binary representation of only the new portion of said representation of the Java class file;

e. sending said binary representation of said new portion to the client device;

f. loading said binary representation of said new portion into said client device; and,

g. copying said binary representation into the internal class structures in the interpreter of Java virtual machine of the client device.

3. The method of Claim 2, wherein step b includes creating a c-code representation of the Java class file and step c includes determining the new portions of said c-code representation, and step d creates a binary representation of only the new portion of said c-code representation.

4. The method of Claim 2, wherein said sending step e includes sending over a wireless network.

5. A system for loading Java class files from a server to a client device comprising:

a. a gateway coupled to said server and responsive to a Java class file for creating a c-code representation of said class file;

b. said gateway creating a binary representation of said c-code representation;

c. a network coupled between said gateway and said client device for sending the binary representation to said client device;

d. a loader for loading said binary representation at said client device; and

e. means for copying said binary representation into the internal class structure in an interpreter of said client device.

6. The system of Claim 5, wherein said gateway includes means for determining new portions of the said c-code representation, and in step b said gateway creates binary representations of only new portions of said c-code representations, and in step c said network sending only said new portions to said client device.

7. A method for loading Java class files to an embedded client device from a server comprising the steps of:

a. gateway retrieving a Java class file;

b. gateway preloading and preresolving the Java class file to produce a representation of the Java class file;

c. determining at the gateway a new portion of the representation;

d. creating at the gateway a binary representation of only said new portion of the preloaded and preresolved representation of the Java class file;

e. sending said binary representation to the embedded client device;

f. loading said binary representation into said embedded client device; and

g. copying said binary representation into the internal class structures in the interpreter of a Java virtual machine of the embedded client device.

8. A system for loading Java class files from a server to an embedded client device comprising:

a. a preloader and preresolver in a gateway coupled to said server and responsive to a Java class file for preloading and preresolving a representation of said class file;

b. said gateway creating a binary representation of said preloaded and preresolved representation of said class file;

c. a wireless network coupled between said gateway and said embedded client device for sending the binary representation to said embedded device;

d. a loader for loading said binary representation at said embedded client device; and,

e. means for copying said binary representation into the internal class structure in an interpreter of said embedded client device.

9. The system of Claim 8, wherein said gateway includes means for determining new portions of said preloaded and preresolved representations of the class and sending only said new portions to said embedded client device.